**UNIVERSIDAD TECNOLÓGICA DEL PERÚ**

****

**FILIAL AREQUIPA**

**CIENCIAS E INGENIERÍAS**

**INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**“Avance de Trabajo Final “**

**Integrante:**

**xxxxxxxxxxxxxx**

**Profesor Asesor:**

**xxxxxxxxxxxxxxxxxxxx**

**AREQUIPA – PERÚ**

**2014**

Contenido

[Introducción 3](#_Toc396406878)

[1.1 Descripción del problema 4](#_Toc396406879)

[1.2 Problema 4](#_Toc396406880)

[1.3 Obtención de los Requerimientos 4](#_Toc396406881)

[1.3.1 Requerimientos funcionales. 4](#_Toc396406882)

[1.3.2 Requerimientos no funcionales. 6](#_Toc396406883)

[1.4 Análisis de requisitos 6](#_Toc396406884)

[1.4.1 Recopilación de Requisitos 6](#_Toc396406885)

[1.4.2 Especificación de Requisitos 6](#_Toc396406886)

[1.4.2.1 Caso de Uso 6](#_Toc396406887)

[1.4.2.2 Historias de Usuario 6](#_Toc396406888)

[1.5 Arquitectura del Software 6](#_Toc396406889)

[1.5.1 Diagramas de Clases 6](#_Toc396406890)

[1.5.2 Diagramas de Base de Datos 6](#_Toc396406891)

[1.5.2.1 Diagrama Lógico de la Base de datos 7](#_Toc396406892)

[1.5.2.2 Script de la Base de Datos 7](#_Toc396406893)

[1.5.2.3 Procedimientos Almacenados 9](#_Toc396406894)

[1.5.2.4 Vistas 11](#_Toc396406895)

[1.5.2.5 Funciones 12](#_Toc396406896)

[1.5.2.6 Triggers 12](#_Toc396406897)

[1.5.3 Diagrama de despliegue 13](#_Toc396406898)

[1.5.4 Diagrama de secuencia 13](#_Toc396406899)

[1.6 Desarrollo de la Aplicación 13](#_Toc396406900)

[1.6.1 Interfaz Gráfica de Usuario 14](#_Toc396406901)

[1.6.2 Hardware Empleado en el Software 17](#_Toc396406902)

[Lector de Código de Barras Symbol Ds 6608 2d 17](#_Toc396406903)

[Arduino Mega Rev 3 2560 18](#_Toc396406904)

[1.6.3 Diagrama de Montaje de Componentes 20](#_Toc396406905)

[1.6.4 Lista de Componentes Electrónicos 21](#_Toc396406906)

[1.6.5 Código Fuente del Hardware Arduino Mega 2560 Rev 3 22](#_Toc396406907)

# Introducción

En el presente trabajo se pretende dar una solución eficaz a problemas que enfrenta el sector privado, para ello se integrara hardware y software para solucionar los problemas dentro delas empresas.

Ocultar y mostrar comentarios en ficha Revisar

http://es.wikipedia.org/wiki/Ingenier%C3%ADa\_de\_software

# Lista de figuras

# 1.1 Descripción del problema

La empresa Bouby sac es una empresa que tiene como actividad principal la construcción y mantenimiento de carreteras, para realizar esta actividad la empresa requiere de grandes cantidades de combustible que deben ser almacenados en campo, para distribuir a los equipos.

La empresa controla la salida de distribución de combustible a los diferentes equipos pesados que realizaran labores de la actividad principal de la empresa, mediante el formato F\_1524 (ISO 9000).

# 1.2 Problema

La empresa ha registrado mayor consumo de combustible durante los últimos meses, esto hace sospechar que existe personal que está hurtando combustible en el área de trabajo, ya que el formato F\_1524 solo es un registro para el ingreso y salida de combustible, pero no registra realmente el momento que se realiza la distribución al equipo, también se presenta el problema del stock actual de combustible, para ello la empresa requiere un software que tenga un registro de la distribución de combustible en el que se debe registrar mediante video y foto el momento de la distribución, además este debe tener un registro del stock actual de combustible y como de seguridad debe controlar mediante sensores la temperatura del depósito para la acción inmediata en caso de incendio.

# 1.3 Obtención de los Requerimientos

## 1.3.1 Requerimientos funcionales.

* El sistema debe permitir registrar los datos de los colaboradores (Operador de equipo pesado) y emitir un fotocheck para poder acceder al depósito de combustible, para que el colaborador pueda abastecer de combustible su vehículo.
* El software debe contar con un usuario administrador que permita otorgar el acceso al depósito de combustible para que el colaborador pueda abastecerse de combustible.
* El abastecimiento de combustible debe poder realizarse mediante un código otorgado por el administrador del sistema al colaborador(Operador de equipo pesado)
* El sistema debe permitir al usuario administrador buscar y consultar la información de ingresos y salida del depósito de combustible por colaborador (Operador de equipo pesado). El sistema debe realizar la captura de video y foto de la distribución de combustible
* El sistema debe cumplir las medidas de seguridad de las instalaciones, es decir no puede haber una distancia menor a 50m desde las instalaciones del servidor a los dispositivos de captura y sensores.
* El sistema debe emitir una alarma en caso de incendio en el depósito de combustible.

## 1.3.2 Requerimientos no funcionales.

* La búsqueda de información no debe tardar mas de cinco segundos.
* Si se supera este plazo, el sistema detiene la búsqueda y muestra los resultados encontrados.
* La captura de los sensores no debe tardar mas de 2 segundos.

# 1.4 Análisis de requisitos

## 1.4.1 Recopilación de Requisitos

## 1.4.2 Especificación de Requisitos

## 1.4.2.1 Caso de Uso

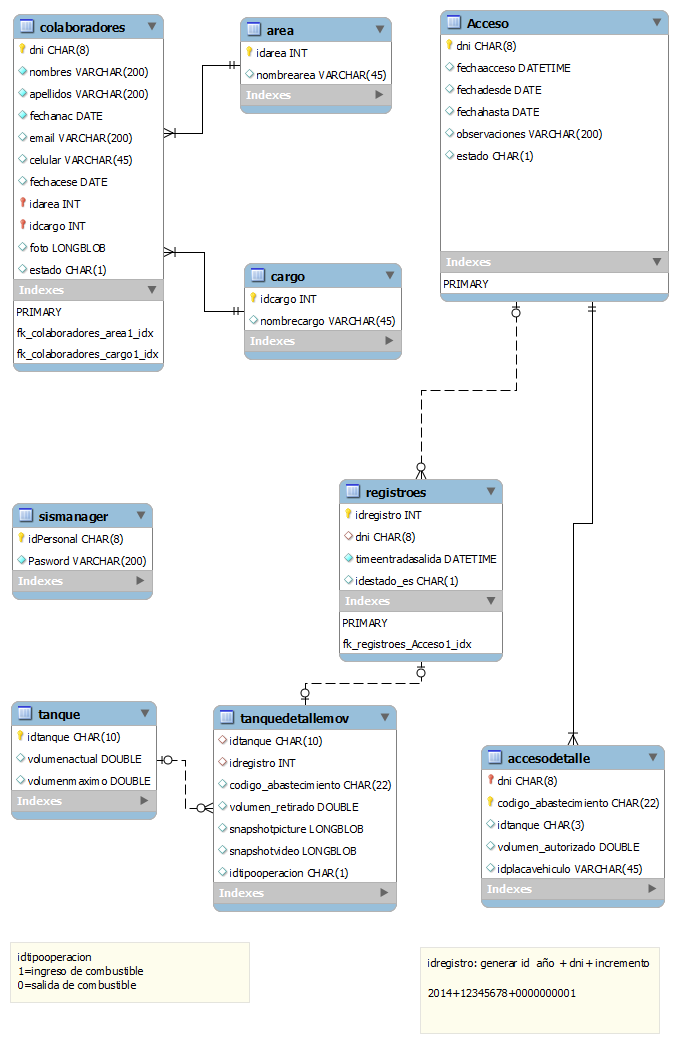
## 1.4.2.2 Historias de Usuario

# 1.5 Arquitectura del Software

## 1.5.1 Diagramas de Clases

## 1.5.2 Diagramas de Base de Datos

## 1.5.2.1 Diagrama Lógico de la Base de datos



## 1.5.2.2 Script de la Base de Datos

CREATE TABLE acceso (

dni varchar2(8),

fechaacceso timestamp,

fechadesde date,

fechahasta date,

observaciones varchar2(200),

estado varchar2(1),

PRIMARY KEY (dni)

);

CREATE TABLE accesodetalle (

dni varchar2(8) ,

codigo\_abastecimiento varchar2(22) ,

idtanque varchar2(3) ,

volumen\_autorizado number(8,2) ,

idplacavehiculo varchar2(45) ,

PRIMARY KEY ( dni , codigo\_abastecimiento )

)

CREATE TABLE area (

idarea number(3),

nombrearea varchar2(45) ,

PRIMARY KEY ( idarea )

)

CREATE TABLE cargo (

idcargo number(3) ,

nombrecargo varchar2(45) ,

PRIMARY KEY ( idcargo )

)

CREATE TABLE colaboradores (

dni varchar2(8) ,

nombres varchar2(200) ,

apellidos varchar2(200) ,

fechanac date ,

email varchar2(200) ,

celular varchar2(45) ,

fechacese date ,

idarea number(3) ,

idcargo number(3) ,

foto blob,

estado varchar2(1) ,

PRIMARY KEY ( dni )

)

CREATE TABLE registroes (

idregistro number(38) ,

dni varchar2(8) ,

timeentradasalida timestamp ,

idestado\_es varchar2(1) ,

PRIMARY KEY ( idregistro )

)

CREATE TABLE sismanager (

idPersonal varchar2(8) ,

Pasword varchar2(200) ,

PRIMARY KEY ( idPersonal )

)

CREATE TABLE tanque (

idtanque varchar2(10) ,

volumenactual number(8,2) ,

volumenmaximo number(8,2) ,

PRIMARY KEY ( idtanque )

)

CREATE TABLE "TANQUEDETALLEMOV"

( "IDTANQUE" VARCHAR2(10),

"IDREGISTRO" NUMBER(38,0),

"CODIGO\_ABASTECIMIENTO" VARCHAR2(22),

"VOLUMEN\_RETIRADO" NUMBER(8,0),

"SNAPSHOTPICTURE" BLOB,

"SNAPSHOTVIDEO" BLOB,

"IDTIPOOPERACION" VARCHAR2(1)

)

## 1.5.2.3 Procedimientos Almacenados

**SP\_NUEVO\_REGISTRO\_ES: permite registrar el ingreso al depósito de combustible.**

create or replace procedure sp\_nuevo\_registro\_es  
(dni IN VARCHAR2,idregistro IN number,  
timeentradasalida IN TIMESTAMP,  
idestado\_es IN VARCHAR2)  
is  
begin  
INSERT INTO registroes  
(idregistro, dni,  
timeentradasalida,  
idestado\_es)  
VALUES  
(idregistro, dni,  
timeentradasalida,  
idestado\_es);  
end sp\_nuevo\_registro\_es;​

**SP\_NUEVO\_COLABORADOR: Registra un nuevo colaborador en la base de datos.**

create or replace procedure sp\_Nuevo\_Colaborador (dni  in varchar2,  
     nombres  in varchar2,  
     apellidos  in varchar2,  
     fechanac  in date,  
     email  in varchar2,  
     celular  in varchar2,  
     fechacese  in date,  
     idarea  in number,  
     idcargo  in number,  
     foto  in blob,  
    estado in varchar2)  
is  
BEGIN  
 INSERT INTO colaboradores  
(dni,  
nombres,  
apellidos,  
fechanac,  
email,  
celular,  
fechacese,  
idarea,  
idcargo,  
foto,  
estado)  
VALUES  
(dni,  
nombres,  
apellidos,  
fechanac,  
email,  
celular,  
fechacese,  
idarea,  
idcargo,  
foto,  
estado);  
END sp\_Nuevo\_Colaborador;​

**SP\_NUEVO\_TANQUE:Nos permite ingresar al sistema un nuevo depósito de combutible.**

create or replace PROCEDURE sp\_Nuevo\_Tanque(idtanque in varchar2, volumenactual in number, volumenmaximo in number)  
is  
BEGIN  
INSERT INTO tanque  
(idtanque,  
volumenactual,  
volumenmaximo)  
VALUES  
(idtanque,  
volumenactual,  
volumenmaximo);  
END  sp\_Nuevo\_Tanque;​

**SP\_NUEVOACCESO: permite otorgar los permisos necesarios para activar el fotocheck del personal.**

create or replace PROCEDURE sp\_NuevoAcceso ( dni in  varchar2,  
     fechaacceso  timestamp,  
     fechadesde  timestamp,  
     fechahasta  timestamp,  
     observaciones  varchar2,  
     estado  varchar2  
    )  
    is  
BEGIN  
INSERT INTO acceso  
(dni,  
fechaacceso,  
fechadesde,  
fechahasta,  
observaciones,  
estado)  
VALUES  
(dni,  
fechaacceso,  
fechadesde,  
fechahasta,  
observaciones,  
estado);  
END sp\_NuevoAcceso;

## 1.5.2.4 Vistas

Lista todas las áreas activas en el servidor.

CREATE OR REPLACE FORCE VIEW  "LISTAR\_AREA" ("IDAREA", "NOMBREAREA")

AS   
  select "IDAREA","NOMBREAREA" from area;​

Lista todos los cargos o puestos de trabajo que tienen la empresa.

CREATE OR REPLACE FORCE VIEW  "LISTAR\_CARGO" ("IDCARGO", "NOMBRECARGO") AS   
  select "IDCARGO","NOMBRECARGO" from cargo;​

/////DEVUELVE UNA CANTIDAD X DE REGISTROS/////

select \* from registroes where ROWNUM <= 2 and dni='41715787' order by idregistro desc;

////DEVUELVE SI EL SERVIDOR SE ENCUENTRA DENTRO DEL RANGO DEFECHA DE ACCESO/////

SELECT \* FROM ACCESO WHERE DNI='11111111' AND (TO\_DATE(SYSDATE) BETWEEN TO\_DATE(FECHADESDE) AND TO\_DATE(FECHAHASTA));

//

## 1.5.2.5 Funciones

**FN\_FECHAYHORADELSERVIDOR: Devuelve la hora del servidor**

create or replace FUNCTION fn\_FechayHoradelServidor()  
RETURN varchar2  
IS  
  result varchar2;  
BEGIN  
   
  
 select systimestamp INTO result from dual;  
  
  return(result);  
EXCEPTION   
WHEN NO\_DATA\_FOUND THEN  
  return "desconocido";  
END;​

## 1.5.2.6 Triggers

**colaboradores\_no\_borrar: permite mantener la integridad de la información de la tabla colaboradores, para que no sea eliminada.**

create or replace TRIGGER  colaboradores\_no\_borrar  
BEFORE  
delete on colaboradores  
for each row  
begin  
RAISE\_APPLICATION\_ERROR (-20500,'No esta permitido borrar registros de la tabla colaboradores');  
  
end;​

**registroes\_no\_borrar: de acuerdo al requerimiento el registro de ingreso debe permanecer en el tiempo sin poder eliminarse.**

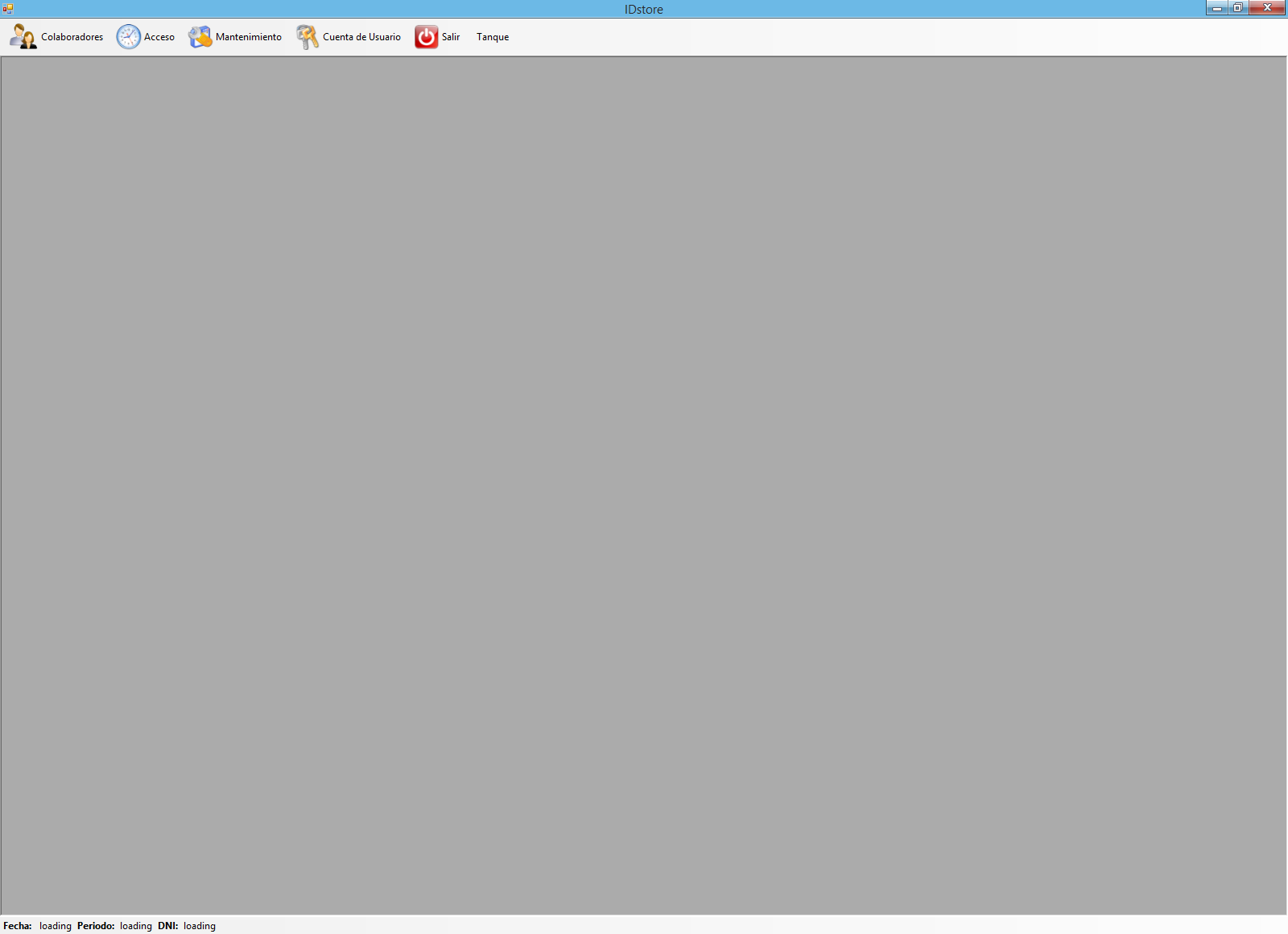
create or replace TRIGGER  registroes\_no\_borrar  
BEFORE  
delete on registroes  
for each row  
begin  
RAISE\_APPLICATION\_ERROR (-20500,'No esta permitido borrar registros de entrada y salida');  
  
end;

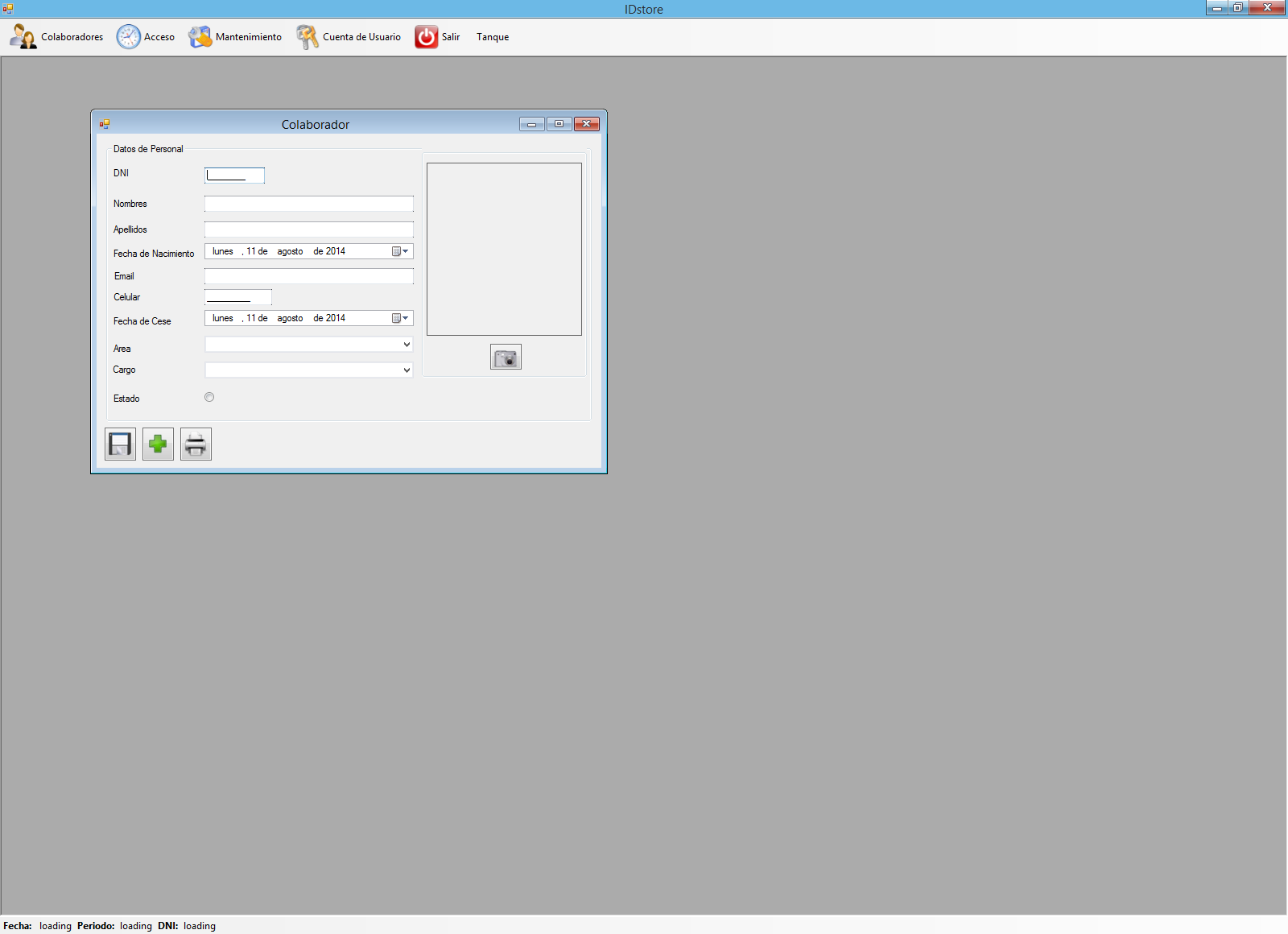
## 1.5.3 Diagrama de despliegue

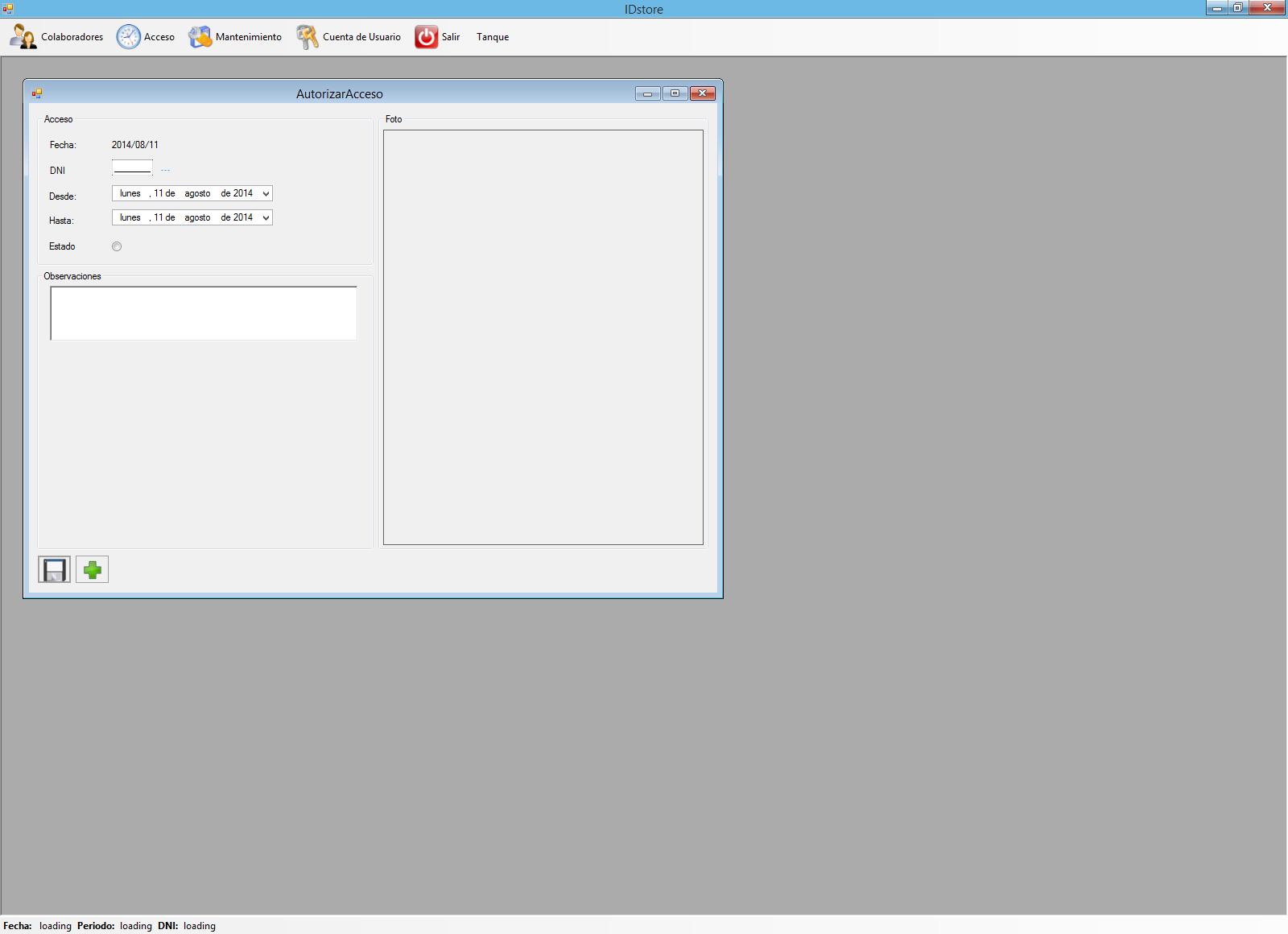
## 1.5.4 Diagrama de secuencia

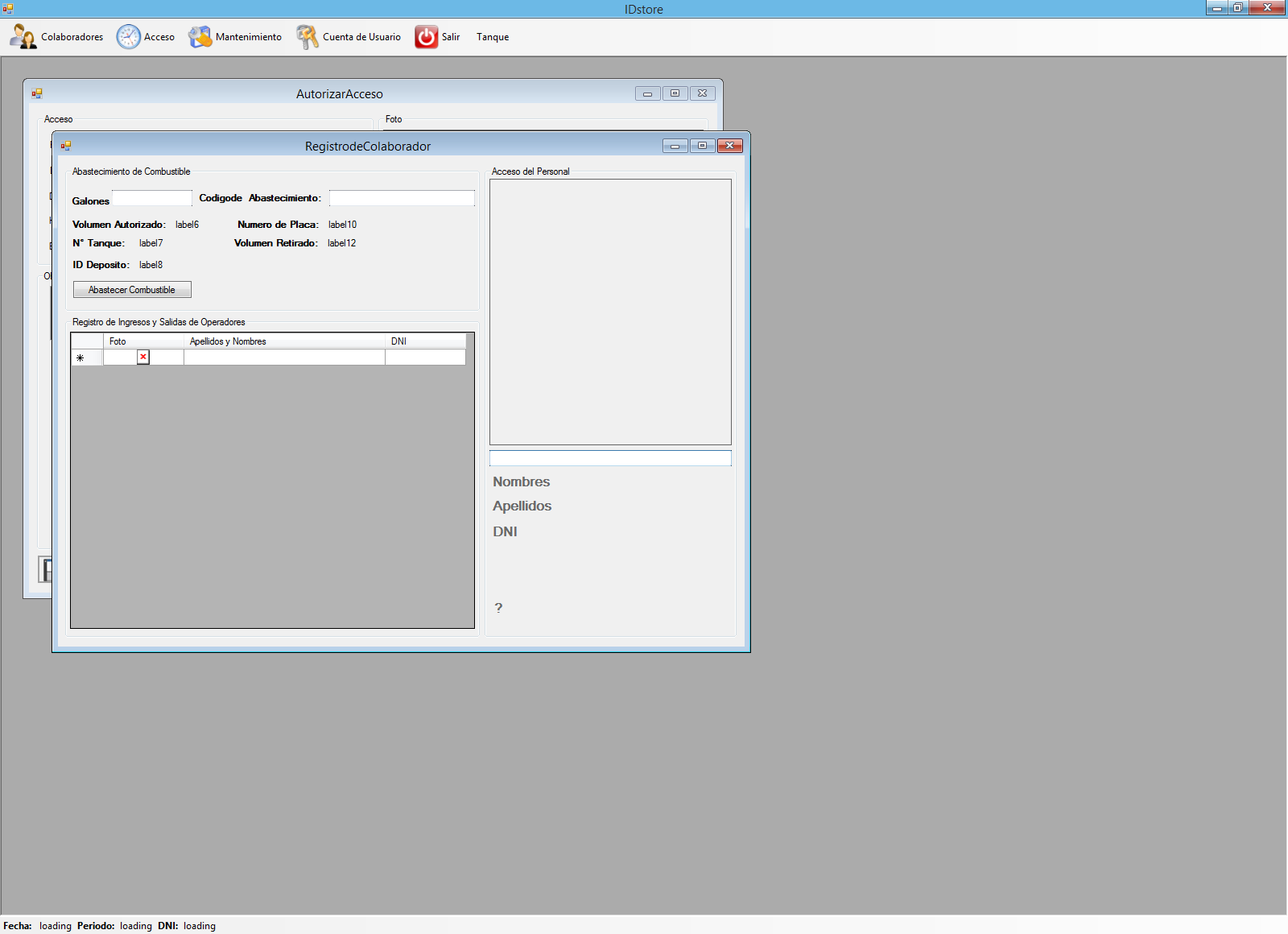
# 1.6 Desarrollo de la Aplicación

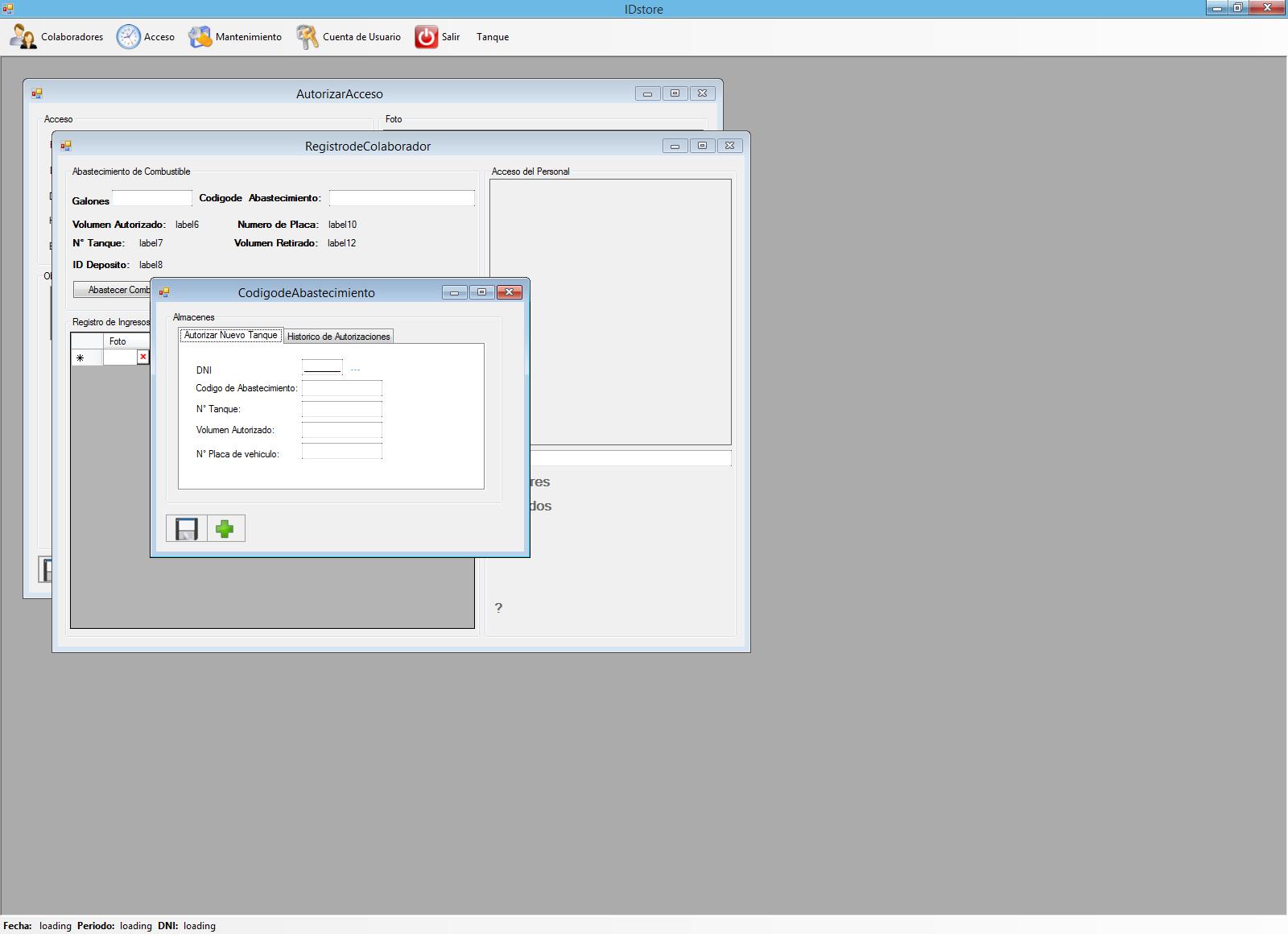
## 1.6.1 Interfaz Gráfica de Usuario

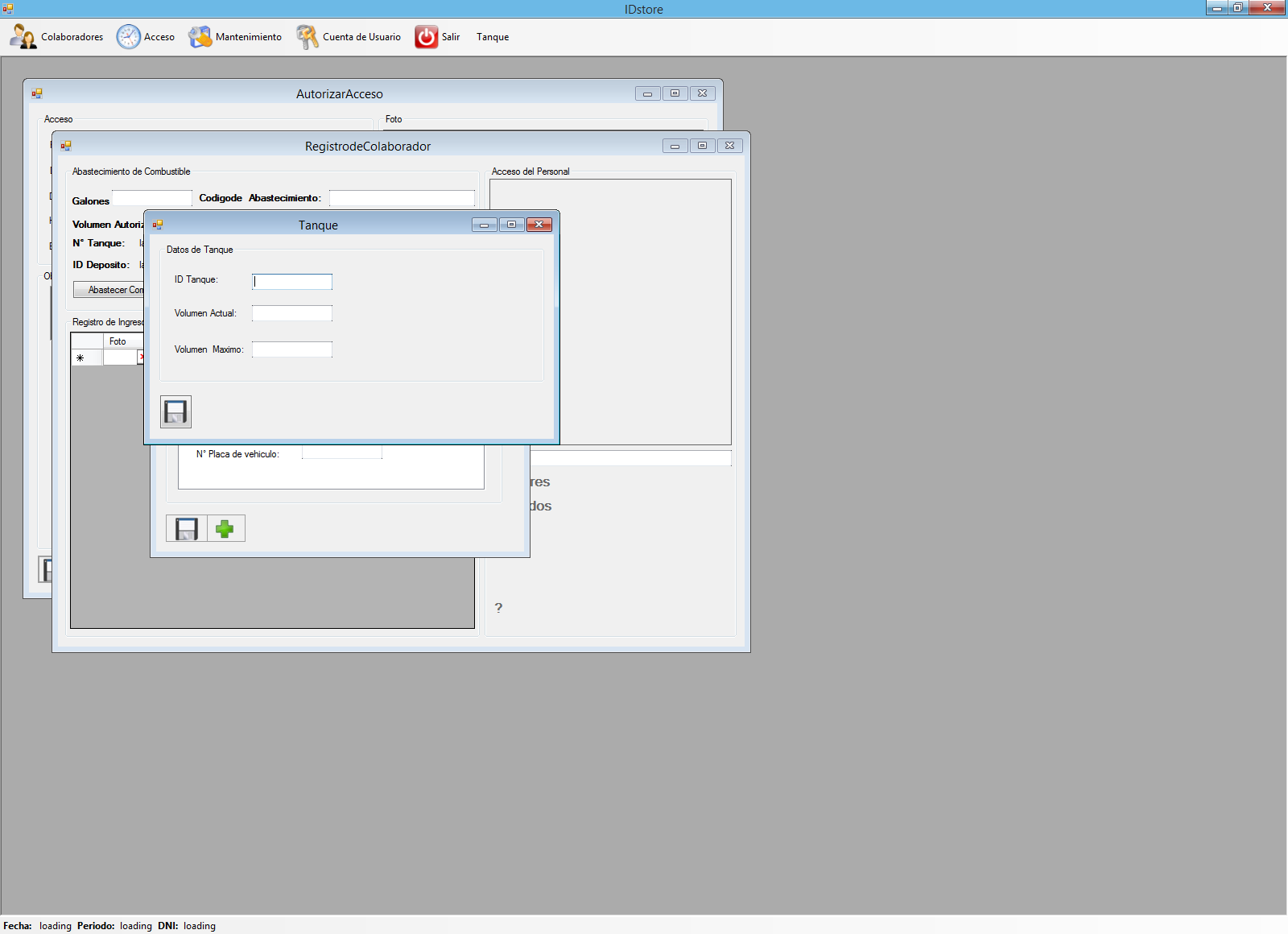












## 1.6.2 Hardware Empleado en el Software

Siguiendo la línea de cumplimiento de objetivos en el hardware, los componentes del hardware empleados en el presente trabajo son:

## Lector de Código de Barras Symbol Ds 6608 2d

Los lectores de código de barras de permiten acelerar los procesos de lectura, alta y control de inventarios en clínicas médicas y hospitales con captura de datos superior y perfecta integración con otros sistemas de redes y de Sistemas de Información integrados con la lectura de códigos de barras.

Para cumplir los objetivos se dio por viable la utilización del hardware Lector de código de Barras (Motorola Symbol DS 6608) por las siguientes razones:

* Agilidad en la lectura de códigos 1D y 2D.
* Ante la pérdida de una Tarjeta de Identificación del personal el lector puede leer el código de barras del DNI, permitiendo su acceso rápidamente.
* La impresión de Tarjeta de Identificación es barato, ya que representaría un gasto de S/ 0.50 CU.
* Porcentaje de erro en la lectura es muy bajo, siendo este de 0.09 %.
* El lector de código de barras es fácil de conectar e instalar, ya que este cuenta con conexión USB.
* No necesita de una toma externa de corriente eléctrica, ya que su funcionamiento es similar el teclado de una computadora.
* El costo del hardware es viable S/ 450.00.



Figura 1 Lector de Código de Barras Symbol Ds 6608 2d.

## Arduino Mega Rev 3 2560

Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida.4 Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

Las razones de su elección en el trabajo son:

* Compatible con los diversos lenguajes de programación (Phyton, Java, C#, Visual Basic, Mono, etc. ).
* El costo del hardware es económicamente viable S/ 150.00.
* El hardware ofrece la posibilidad de controlar múltiples dispositivos electrónicos: sensores de temperatura, motores, wifi, Ethernet, etc.

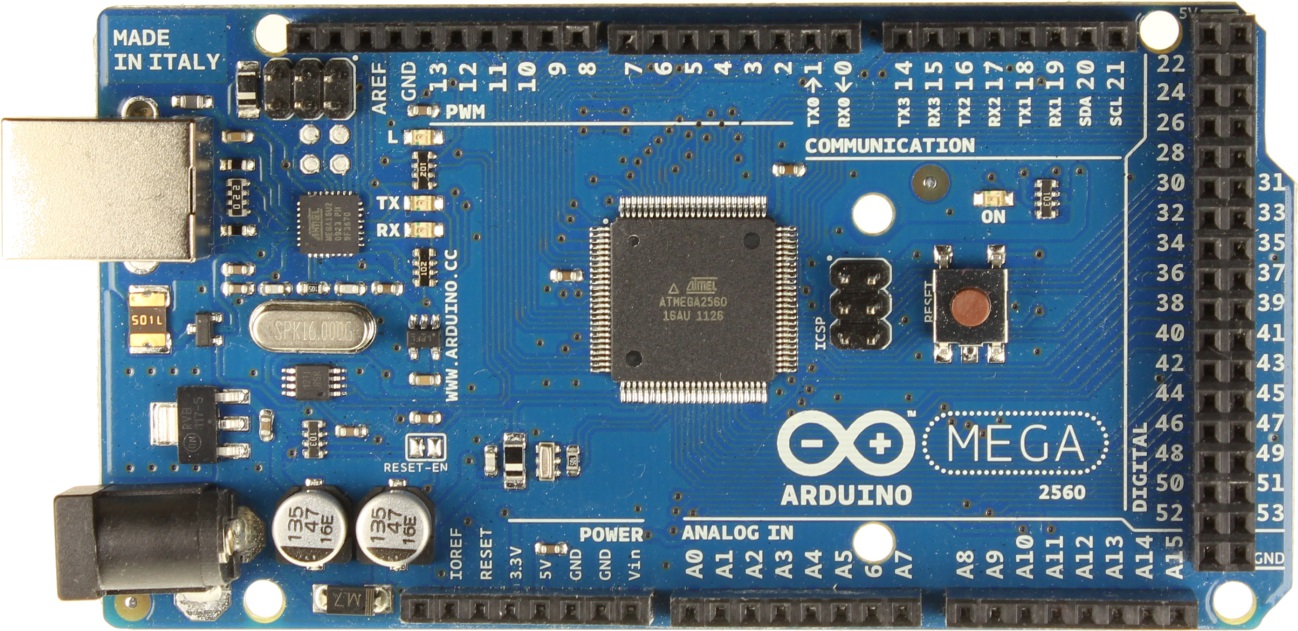


Figura 2 Arduino Mega Rev 3 2560.

## 1.6.3 Diagrama de Montaje de Componentes

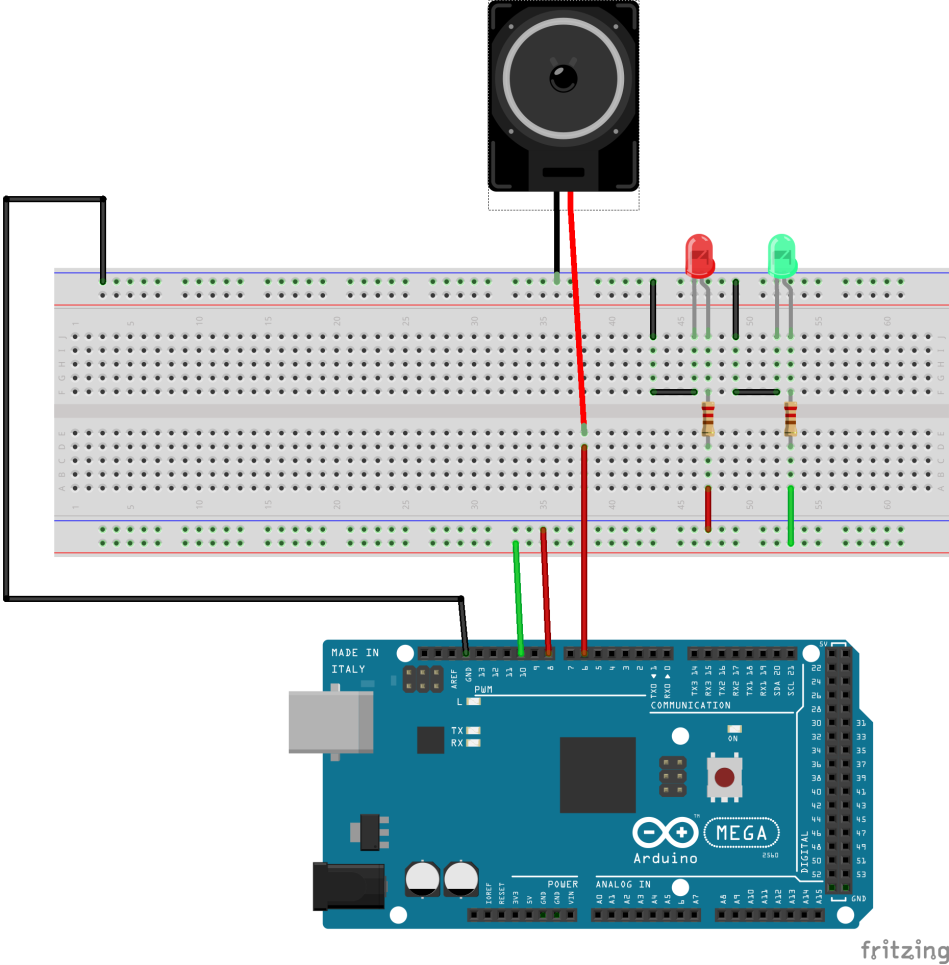


Figura 3 Diagrama de componentes electrónicos IDCheck.

## 1.6.4 Lista de Componentes Electrónicos

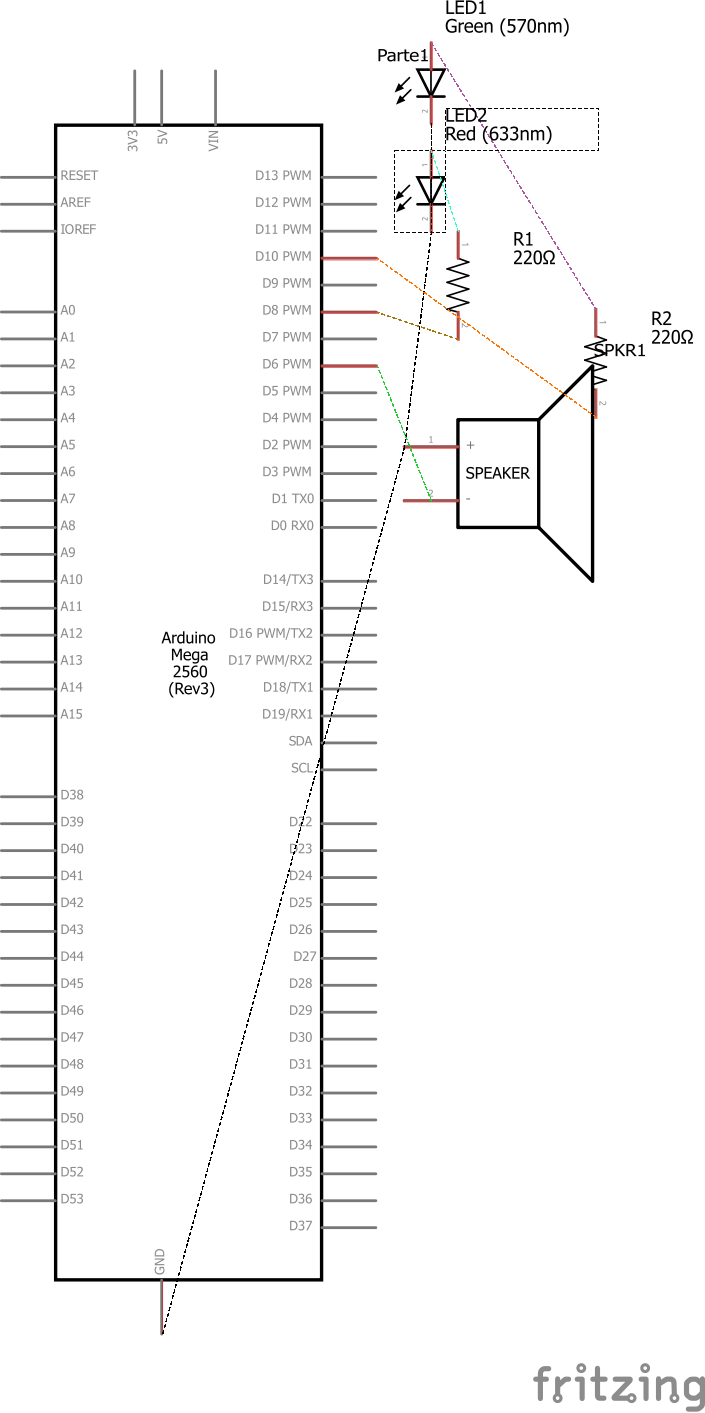


Figura 4 Vista de valores de componentes electrónicos.

## 1.6.5 Código Fuente del Hardware Arduino Mega 2560 Rev 3

